

Subcellular behavior model enables highly precise temporal super-resolved live-cell imaging

Bo Yan

b.yan@fudan.edu.cn

Fudan University <https://orcid.org/0000-0001-5692-3486>

Zixian Zhang

Fudan University

Ruian He

Fudan University

Weimin Tan

School of Computer Science, Shanghai Key Laboratory of Intelligent Information Processing, Fudan University, Shanghai <https://orcid.org/0000-0001-7677-4772>

Xinda Chen

Fudan University

Article

Keywords:

Posted Date: May 23rd, 2025

DOI: <https://doi.org/10.21203/rs.3.rs-6612059/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: There is **NO** Competing Interest.

To facilitate biologists in using this advanced artificial intelligence tool to accelerate scientific discoveries, we have made our code and documentation open source on GitHub (<https://github.com/sduzzx857/SuB-VFI>). The data used in this work are available in the Zenodo repository (<https://zenodo.org/records/14043236>), and the pretrained weights can be accessed on

Google Drive

(https://drive.google.com/drive/folders/1v3XXAmuJZOov1JaOiWMNhmJvKj8iM_Ee).

Additionally, we have provided a live demo for SuB-VFI on Colab

(<https://colab.research.google.com/github/sduzzx857/SuB-VFI/blob/main/demo.ipynb>).

Subcellular behavior model enables highly precise temporal super-resolved live-cell imaging

Zixian Zhang^{1†}, Ruian He^{1†}, Weimin Tan^{1*}, Xinda Chen¹,
Bo Yan^{1*}

¹School of Computer Science, Shanghai Key Laboratory of Intelligent
Information Processing, Fudan University, Shanghai, 200438, China.

*Corresponding author(s). E-mail(s): wmtan@fudan.edu.cn;
byan@fudan.edu.cn;

[†]These authors contributed equally to this work.

Abstract

Live-cell imaging is essential for studying dynamic cellular processes but is limited by temporal resolution constraints due to photobleaching and phototoxicity. Video frame interpolation techniques have been applied to address this. However, existing pixel-level methods are insufficient in capturing subcellular dynamics, as they focus mainly on appearance features while neglecting other subcellular characteristics. Here, we model the behavior of subcellular particles and combine behavioral features with appearance features. This approach enhances the accuracy of particle matching and improves interpolation outcomes. We tested our approach on simulated and real datasets and evaluated its performance in downstream particle tracking tasks. Compared to other interpolation methods, our approach effectively handles a variety of challenging scenarios, such as the sudden appearance or disappearance of particles, particle overlap, and motion pattern changes. Consequently, our method offers a novel approach to improve the temporal resolution of subcellular imaging and reduce sample exposure time and phototoxicity.

1 Introduction

With the significant advancements in microscopy imaging technologies, the demands for studying rapid dynamic processes within living cells have also escalated. This requires both high imaging acquisition speed and quality. However, achieving both simultaneously is often a trade-off: higher image quality, such as improved signal-to-noise ratio (SNR) and image resolution, typically comes at the cost of reduced acquisition speed; conversely, higher speed constrains the contextual information in the images. Additionally, prolonged exposure times can compromise the health of the sample, further complicating efforts to achieve high acquisition speeds. Therefore, it is essential to explore methods for enhancing the temporal resolution of image acquisition while maintaining image quality.

Deep learning (DL) algorithms have been utilized in microscopy image post-processing for several years, yielding groundbreaking advancements across various applications[1–12]. Among these algorithms, video frame interpolation (VFI) has emerged as a crucial low-level computer vision task, aimed at generating non-existent intermediate frames between consecutive inputs. This approach significantly enhances the temporal resolution of video sequences, offering a valuable tool for capturing dynamic processes within living cells at higher temporal detail[11, 12]. CAFI[12] has attempted to apply this technology in subcellular living-cell imaging, testing two deep learning-based video frame interpolation (VFI) methods, ZS[13, 14] and DAIN[15], demonstrating that they significantly outperform traditional interpolation methods.

However, CAFI[12] noted some limitations of deep learning-based VFI methods, such as artifacts and difficulty handling fast-moving objects[16], which can impact the quality of interpolation results. Our investigation reveals that current interpolation methods, whether reliant on pixel-level motion estimation[17–20] or convolutional operations[21–23], predominantly focus on subcellular appearance features. While appearance features are adequate for matching in natural images, subcellular components often exhibit very similar appearance characteristics, such as particles with closely resembling shapes and brightness levels. Consequently, relying solely on appearance features proves inadequate for effectively distinguishing and matching subcellular components of the particle type. This highlights the necessity for simultaneous analysis of multiple channels to provide a better description of particles for identifying subcellular dynamics.

Subcellular components typically exhibit a variety of behaviors, such as substance transport[24–26] and signal transduction[27, 28]. In recent years, researchers have modeled and analyzed subcellular behaviors, achieving significant results[29–32]. We focused on the behavior of particles within subcellular images, where different behaviors usually manifest distinct motion patterns. We reasoned that modeling the behavior of particles would aid in better identifying the motion of each particle, thereby assisting interpolation models in distinguishing and matching particles across consecutive frames.

Here, we devised a subcellular behavior-based video frame interpolation (SuB-VFI) model to achieve more accurate interpolation predictions (Fig.1a). By extracting the group behavioral feature for each particle (Fig.1b) and effectively integrating it with

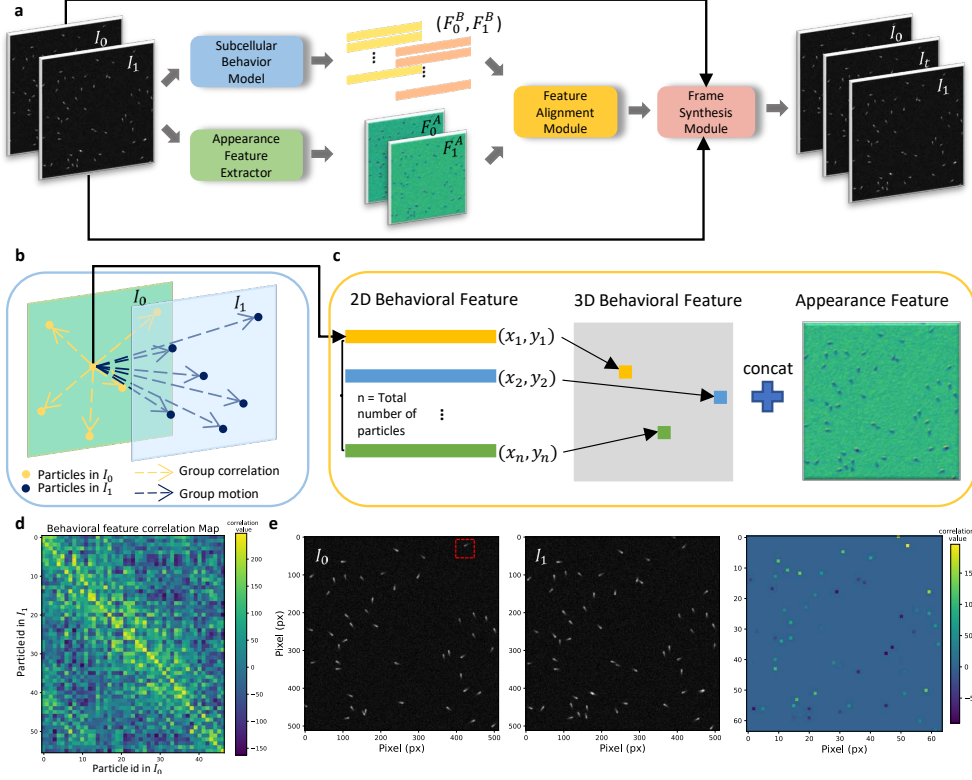


Fig. 1 Subcellular behavior-based VFI model. **a**, Architecture of our subcellular behavior-based VFI model, which comprises a Subcellular Behavior Model, Appearance Feature Extractor, Feature Alignment Module, and Frame Synthesis Module. I_0 , I_1 , and I_t represent the input of two consecutive frames and the output of the intermediate frame, while F^B and F^A denote the behavioral features and appearance features. **b**, Subcellular Behavior Model: Group behavioral features are extracted by modeling the behavior of particles within two frames I_0 and I_1 . **c**, The Feature Alignment Module maps behavioral features from two dimensions to three dimensions, integrating them with appearance features. (x_i, y_i) represents the coordinates of the particle. **d**, Correlation map for all pairs of behavioral features of particles within the frames I_0 and I_1 . **e**, Frames I_0 (left, 512×512 pixels), I_1 (middle, 512×512 pixels), and resized correlation image (right, 64×64 pixels) between particle P in the frame I_0 (highlighted in red box) and all particles within the frame I_1 .

the appearance feature (Fig.1c), our approach significantly enhances both the distinctiveness of particles and the inter-frame correlation between corresponding particles (Fig.1d, 1e). This behavioral feature effectively addresses the limitations of appearance features in terms of particle recognition.

We validated the superior performance of Sub-VFI against existing VFI methods using both simulated particle datasets from the ISBI Particle Tracking Challenge[33] and real experimental datasets we collected. The simulated dataset includes three motion patterns: directed motion, Brownian motion, and random switching motion between these two. These motion patterns are representative of various subcellular

scenarios. We collected three real datasets from the literature, specifically from APC-binding protein EB1[31], CC chemokine receptor 5 (CCR5)[32], and lysosomes[12]. These real datasets come from different biological samples and encompass a range of representative subcellular behaviors, with motion patterns that correspond directly to directed motion, Brownian motion, and random switching motion. Across these datasets, our approach consistently outperforms or matches the best-performing methods, demonstrating exceptional robustness. Specifically, Sub-VFI delivers interpolation results that align more closely with ground truth, especially in challenging scenarios such as sudden particle appearances or disappearances, large particle displacements, and particle overlap—issues that are common and difficult to address in subcellular research[34–37], where other methods often struggle. In the simulated datasets, our approach surpassed other methods, achieving a PSNR enhancement of up to 0.28 dB and an SSIM improvement of up to 0.42%. In the real datasets, it resulted in a PSNR increase of up to 0.14 dB and an SSIM enhancement of up to 0.82%. Extensive tracking analysis on both simulated and real data further validates the potential of our approach, which produces interpolated sequences with fewer trajectory errors, including interruptions, incorrect connections, and missing trajectories. Our method also achieved a leading position across the five tracking metrics in the ISBI particle tracking challenge[33]. To sum up, leveraging the effectiveness of behavioral features, we offer a novel approach to enhancing the temporal resolution of live-cell imaging without increasing sample health risks, yielding more accurate results that assist researchers in accurately exploring the rapid subcellular dynamics.

2 Results

2.1 Behavioral features are more effective than appearance features in reflecting particle motion patterns

Our objective is to enhance the model’s ability to identify particle motion patterns by incorporating behavioral features; thus, we aim to demonstrate that behavioral features are more effective than appearance features in accurately reflecting these patterns. To validate this, we conducted experiments on both simulated and real datasets, mixing particles exhibiting directed motion and Brownian motion, and extracting their respective behavioral and appearance features. We calculated the correlation of the two types of particle features between consecutive frames to establish the correspondence of particles in these frames. The vector composed of the coordinates of corresponding particles is referred to as the particle-level flow. We subsequently classified the mixed particles based on the average particle-level flow along their trajectories.

Results from both the simulated and real datasets indicate that the classification outcomes using behavioral features closely align with the ground truth (Fig.2a), achieving macro F1-scores of 0.98 and 0.92, respectively. In contrast, the macro F1-scores for appearance features were only 0.88 and 0.74. Due to the visual similarity among particles, the use of appearance features on the simulated dataset led to misclassification of some directed motion particles as Brownian motion particles, while in the real dataset, some Brownian motion particles were incorrectly classified as directed motion. This underscores the greater accuracy of features extracted by the behavioral

122 model in identifying particle motion patterns, thereby aiding the interpolation model
 123 in more effectively matching particles.

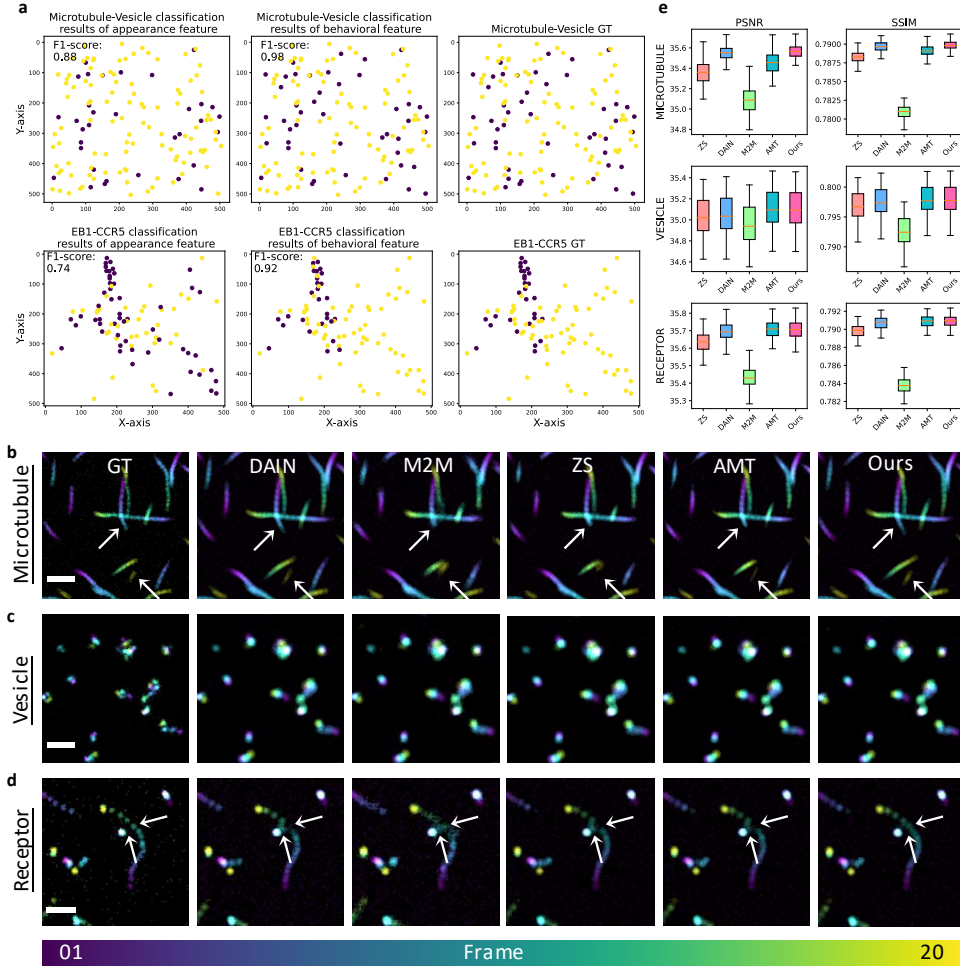


Fig. 2 Quantitative and qualitative assessment of Sub-VFI in simulated particle frame interpolation. **a**, Particle classification results of appearance feature (left), behavioral feature (middle), and ground truth (right) on Microtubule (purple) -Vesicle (yellow) dataset (top) and EB1 (purple) -CCR5 (yellow) dataset (bottom). **b-d**, Temporal color-coded overlaid projection of image sequences of the ground truth and different interpolation methods for Microtubule (**b**), Vesicle (**c**), and Receptor (**d**). White arrows highlight regions of interest for comparing interpolation performance. In the Microtubule dataset, arrows highlight the areas where particles appear and disappear, while in the Receptor dataset, regions of particle absence or artifacts are marked. Scale bar, 30, 30, 18 pixels. **e**, Statistical comparison on the three simulated datasets in terms of PSNR↑/SSIM↑(n = 98 interpolated image frames).

2.2 Behavioral features enable high accuracy in particle matching

To assess the ability of Sub-VFI to predict subcellular dynamics, we first conducted evaluations on simulated datasets from the 2014 ISBI Particle Tracking Challenge[33]. Simulated datasets not only provide a rich source of training data but also offer ground truth tracking, which is essential for testing tracking performance. Furthermore, the simulated datasets encompass a range of particle densities, thereby enhancing the comprehensiveness of the test results. The simulated particle motions include directed motion, Brownian motion, and random switching between these two patterns, as outlined in the challenge. These three motion patterns correspond to the Microtubule dataset, Vesicle dataset, and Receptor dataset, respectively. For each motion pattern, we generated 6,000 images using the ICY data generator[38] as our training set and utilized the dataset downloaded from the challenge as the test set. We compared our predictions against four deep learning-based video frame interpolation (VFI) methods: AMT[17], M2M[18], DAIN[15], and ZS[13, 14]. Information regarding these four interpolation methods and the model training process can be found in the [Methods](#) section. The quality of the interpolated images was assessed using two widely recognized objective pixel-based quality metrics: Structural Similarity Index[39] (SSIM) and Peak Signal-to-Noise Ratio[40] (PSNR).

We first evaluated the efficacy of our approach in addressing the subcellular frame interpolation challenge under low particle density conditions. To do this, we grouped every three consecutive images, using the middle frame as the ground truth. Our observations indicate that our approach effectively inferred the middle frame across various motion types (Fig.2b-2d). In the Microtubule dataset, our approach produced interpolation results closer to the ground truth, particularly during instances of particle appearance or disappearance, when compared to other VFI methods such as M2M, ZS, and AMT. This indicates that our method is better equipped to handle particle occlusion, where a particle appears in only one of the two frames. We attribute this capability to the behavioral model’s ability to capture the motion patterns of particles; even in instances where particles cannot be matched, the model can still predict their positions in the next or previous frame based on their motion patterns. Additional examples of particle occlusion can be found in Supplementary Figure S2. In the Vesicle dataset, due to the relatively small particle movement and the overly smooth interpolation results across all methods, no significant differences were observed between the approaches. Conversely, in the Receptor dataset, where particle movements became larger, the behavior-based VFI approach generated high-quality interpolated frames, while other methods exhibited artifacts or interruptions in particle trajectories. Interruptions in particle trajectories signify instances where the interpolation model failed to accurately match particles, resulting in missing particles in the generated images. Therefore, the experimental results on the Receptor dataset demonstrate that our model excels in particle matching performance.

We also quantified the interpolation accuracy using PSNR and SSIM metrics (Fig.2e). While other methods exhibited variability in performance across different motion patterns, such as DAIN showing a decline in performance on the Vesicle dataset

and AMT on the Microtubule dataset, our approach consistently yielded more stable and superior results. Quantitative analysis further indicates that the behavioral feature exhibits robustness across different motion patterns of particles.

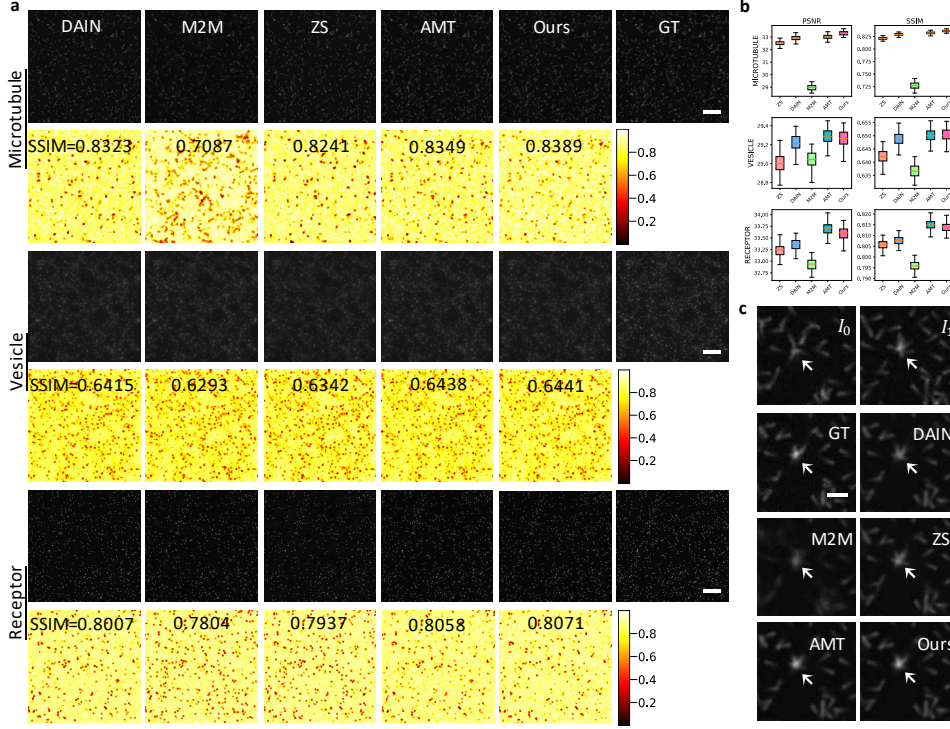


Fig. 3 Quantitative and qualitative assessment in the high-density scenario. **a**, Interpolated frames (top) and SSIM maps (bottom) of GT and different interpolation methods. The images from the Receptor dataset were enhanced. Scale bar, 80 pixels. **b**, Statistical comparison on the three simulated datasets in terms of PSNR $\uparrow$ /SSIM $\uparrow$ ($n = 98$ interpolated image frames). **c**, Examples of input images (I_0 , I_1), ground truth images, and interpolation results from different methods in cases of particle overlap. The overlapping regions are highlighted with white arrows. Scale bar, 20 pixels.

2.3 Behavioral features remain effective in high-density scenarios

Sub-VFI also demonstrates superior interpolation performance in high particle-density images. Particle density details are available in Supplementary Table S1. As particle density increases, the proximity between particles decreases, and overlaps may occur, making it more challenging to distinguish particles across consecutive frames. This significantly complicates the prediction of subcellular dynamics. High-density scenarios are common in real-world datasets, making it crucial to evaluate subcellular structure prediction under these conditions.

We applied our approach to high-density datasets while maintaining the same experimental setup used for the low-density case, and we analyzed the quality of the generated intermediate frames (Fig.3a). In the Microtubule dataset, where particles are larger and more prone to overlap, our approach performed better than other interpolation methods, especially in regions of high particle density. This is evident from the SSIM maps: red regions, indicating lower scores, typically appear in areas of higher particle density due to the increased difficulty in these regions. In the SSIM maps derived from our model’s interpolation results, there are fewer red regions or they are lighter in color. This reflects that the behavioral feature enhances matching accuracy, leading to a reduction in matching errors in high-density areas. Additionally, we observed that when particles overlapped, the brightness in the overlapping regions increased, and our behavior-base VFI model was more effective at capturing this compared to other interpolation techniques (Fig.3c). This indicates that similar to the cases of particle appearance or disappearance in low-density scenarios, the behavior model’s predictions of particle motion enable our method to better handle occlusions such as particle overlaps. On the Vesicle dataset, all methods exhibited a significant decline in performance compared to the low-density scenario. This indicates that the unpredictability of Brownian motion greatly increases the matching difficulty for interpolation models in high-density situations. In the Vesicle and Receptor datasets, our approach performed similarly to AMT. The same conclusion could also be drawn from the PSNR/SSIM results obtained from the high-density dataset (Fig.3b), and these PSNR results were significantly lower than those from the low-density dataset, indicating the increased difficulty of interpolation in high-density particle images.

2.4 Behavioral features enhance particle tracking accuracy

Single particle tracking (SPT) serves as a fundamental tool for quantitatively analyzing dynamic intracellular processes, yet its accuracy can be limited by the temporal resolution of imaging data. Thus, SPT can be considered a downstream task of subcellular structure image interpolation.

To evaluate the quality of interpolated image sequences generated by different methods in the context of SPT, we conducted tracking experiments on the low-density dataset using the TrackMate[41] plugin within the Fiji[42] image analysis platform. For tracking quality assessment, we employed five performance criteria established in the ISBI particle tracking challenge. We compared the particle tracks obtained from the predictions of various methods against the actual ground truth positions. Our approach yielded significantly fewer trajectory interruptions and errors compared to other methods, resulting in interpolated trajectories that align more closely with the ground truth (Fig.4a-4c). In the Microtubule dataset, M2M, ZS, and AMT exhibited instances of late starts or premature ends in their trajectories, consistent with our analysis in low-density scenarios (Fig.2b). Our method, due to its enhanced modeling capability for particle motion patterns, produced trajectories that aligned more closely with the ground truth. In the Vesicle dataset, the trajectories obtained by various methods were similar; however, when particles were close to each other, other methods tended to incorrectly link trajectories. In the Receptor dataset, as the particle movement distance increased, other methods frequently encountered trajectory interruptions.

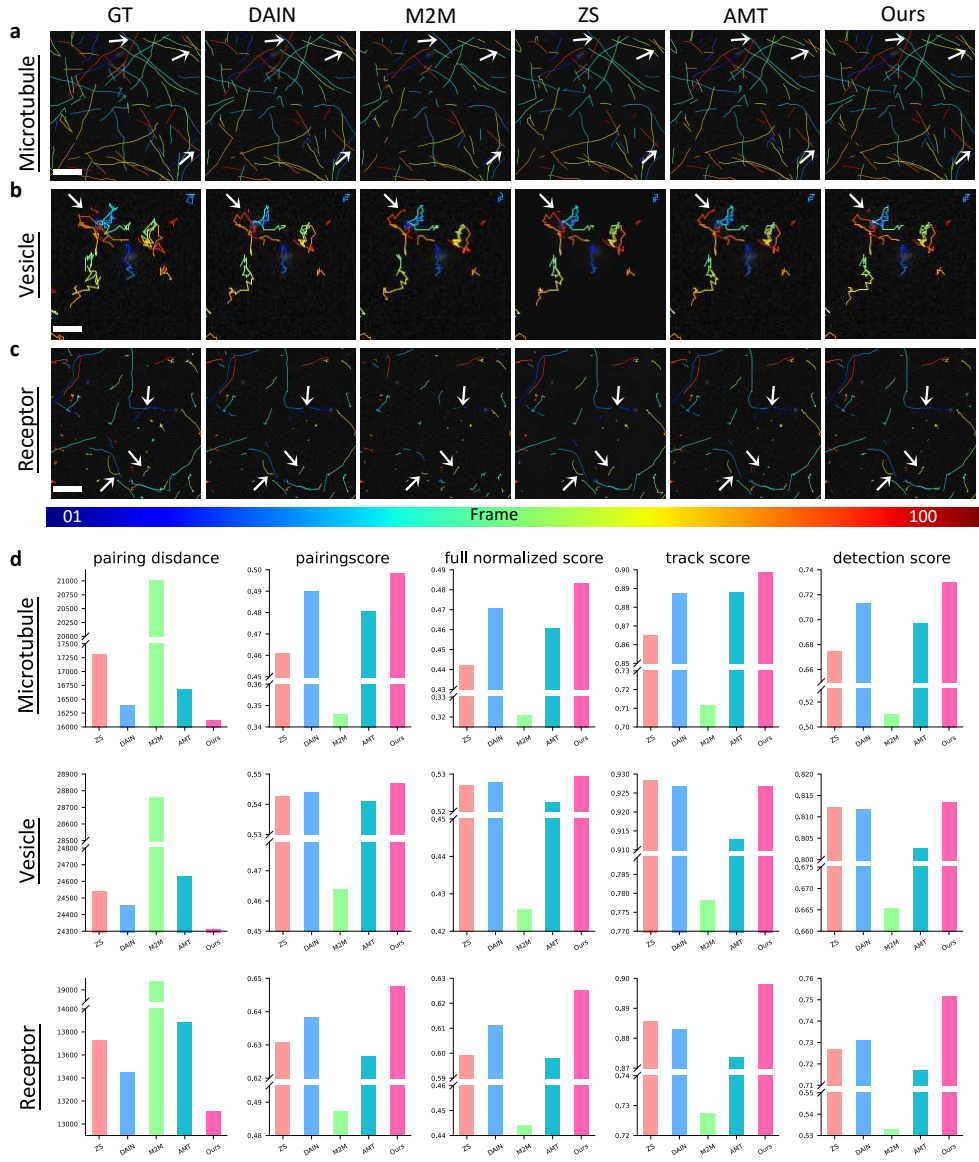


Fig. 4 Quantitative and qualitative assessment of tracking performance on simulated data. **a-c**, TrackMate tracks of interpolation compared with ground truth tracks for Microtubule (**a**), Vesicle (**b**) and Receptor (**c**). White arrows highlight potential error points along trajectories. In the Microtubule dataset, arrows indicate examples of trajectory beginnings and endings; in the Vesicle and Receptor datasets, instances of erroneous connections and trajectory interruptions are highlighted. Scale bar, 30, 15, 30 pixels. **d**, Five tracking evaluation metric results of simulated datasets with ZS, DAIN, M2M, AMT, and Ours interpolation. A smaller pairing distance signifies greater proximity to the ground truth, while the opposite holds for other metrics.

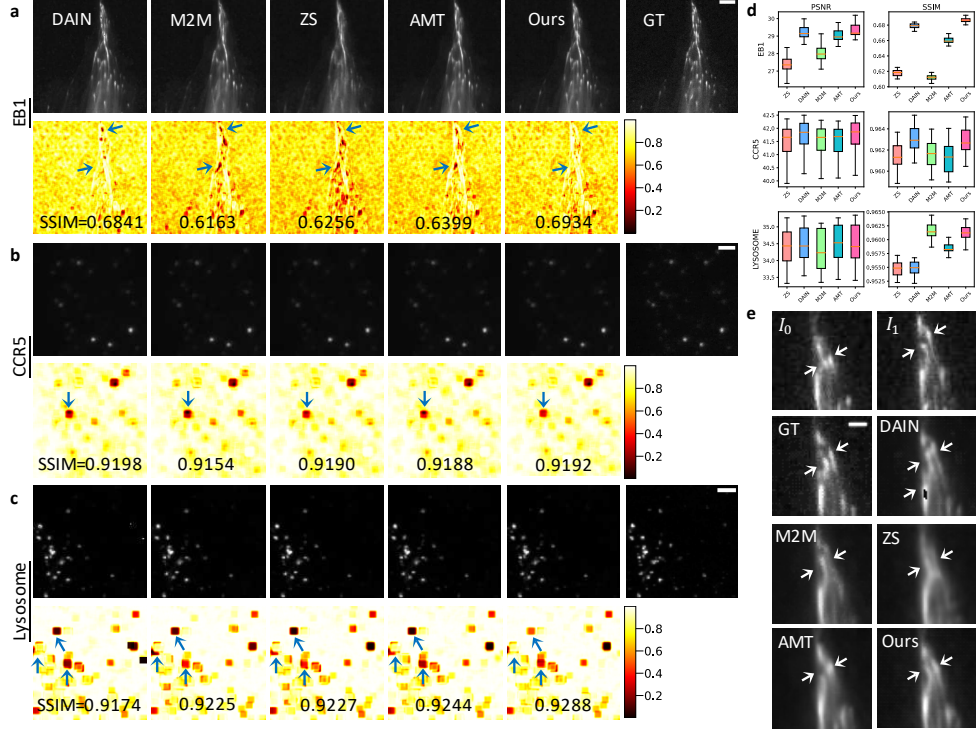


Fig. 5 Quantitative and qualitative assessment in real-data interpolation. **a-c**, Interpolated frames (top) and SSIM maps (bottom) of GT and different interpolation methods. Blue arrows highlight areas where other methods show significant deviations from the ground truth, while our approach does not. Scale bar, 40, 20, 20 pixels. **d**, Statistical comparison on the three real datasets in terms of PSNR $\uparrow$ /SSIM $\uparrow$ ($n = 32, 48, 17$ interpolated image frames from top to bottom). **e**, An example comparison between the interpolation results of different methods and the ground truth on the EB1 dataset, with white arrows highlighting regions of interest. I_0 and I_1 represent input images. Scale bar, 10 pixels.

This also aligned with our analysis in low-density scenarios (Fig.2d). However, our approach reduced errors in erroneous connections and trajectory interruptions, owing to its precise particle-matching ability. Quantitative analysis of tracking performance confirmed the superiority of our approach (Fig.4d), as evidenced by its leading performance across five tracking evaluation metrics in the Microtubule and Receptor datasets, and in most metrics of the Vesicle dataset. Additionally, although M2M performed slightly less well in image quality generation compared to other methods, its tracking performance suffered significantly.

2.5 Behavioral features improve interpolation performance on real datasets

To demonstrate the efficiency of the behavioral feature in handling real experimental data, we conducted tests on three collected real datasets: the EB1 dataset[31], CCR5

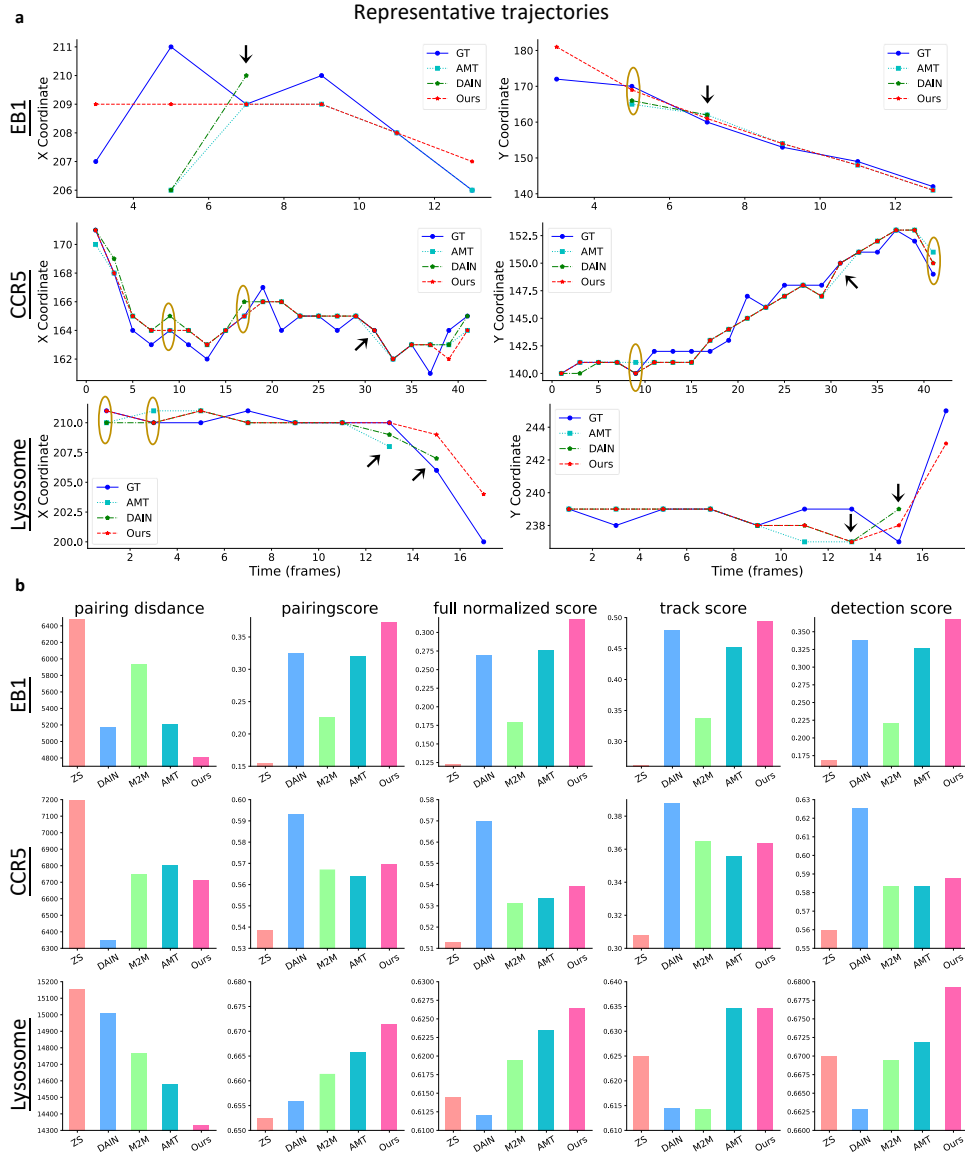


Fig. 6 Quantitative and qualitative assessment of tracking performance on real data. **a**, TrackMate tracks of interpolation compared with ground truth tracks for EB1 (top), CCR5 (middle) and Lysosome (bottom). The black arrows indicate instances where trajectory interruptions occur, while the ellipses highlight areas where our results demonstrate greater accuracy. **b**, Five tracking evaluation metric results of real datasets with ZS, DAIN, M2M, AMT, and Ours interpolation. A smaller pairing distance signifies greater proximity to the ground truth, while the opposite holds for other metrics.

dataset[32], and Lysosome dataset[12]. These datasets correspond to directed motion, Brownian motion, and random switching motion, respectively. We split each dataset into training and test sets at a 1:1 ratio. The training sets were used to fine-tune the pre-trained models of each interpolation method, and interpolation performance was then evaluated on the test sets. Detailed information about the three datasets can be found in Supplementary Table S2.

In our evaluations on real datasets, Sub-VFI exhibited stable interpolation results with higher SSIM. In the EB1 dataset, particles in the upper half of the images become crowded, leading to increased density. Other methods displayed significant errors in this region (indicated by deep red areas in the SSIM maps), while our results exhibited minimal severe errors (Fig.5a), demonstrating the effectiveness of our approach in high-density scenarios. In the CCR5 dataset, the differences among methods were relatively minor due to the small amplitude of particle motion (Fig.5b). In the Lysosome dataset, our results showed fewer significant errors (depicted as deep red regions in the SSIM maps, Fig.5c), confirming that the behavioral feature enables more accurate particle matching in the interpolation model. Quantitatively, our approach consistently maintains a high level of performance across all three datasets (Fig.5d), while other methods exhibited substantial variability. For instance, DAIN performed comparably to our approach on the CCR5 and Microtubule datasets, but its performance significantly declined on the Lysosome dataset. Similarly, AMT achieved high PSNR values but lower SSIM scores. Qualitatively, we further compared the results of various methods on the EB1 dataset. When two particles were in close proximity, M2M, ZS, and AMT lost one of the particles. Although DAIN did not lose any particles, our method produced results that were closer to the ground truth. (Fig.5e).

2.6 Behavioral features reduce trajectory tracking errors on real datasets

We further benchmarked the SPT performance of our approach on real data. Since no ground truth tracking is available for the real datasets, we used the TrackMate plugin[41] to track particles in the test images and treated these tracks as the ground truth for tracking experiments on real data.

We selected a representative trajectory from each dataset for analysis and focused on the results from three well-performing methods: DAIN, AMT, and our approach (Fig.6a). Visualizations of each trajectory can be found in Supplementary Figure S3. In the EB1 dataset, the selected trajectory predominantly moved along the y-axis. Notably, both AMT and DAIN displayed delayed trajectory emergence, whereas the behavioral model’s ability to capture particle motion patterns allowed our method to predict the initial phase of the trajectory, which remained consistent with the subsequent particle movement. Additionally, DAIN exhibited instances of premature trajectory interruption. In the CCR5 dataset, the differences among methods were minimal. In the Lysosome dataset, the selected trajectory first displayed Brownian motion before transitioning to directed motion. During the Brownian motion phase, the results from various methods were similar; however, as particles shifted from Brownian to directed motion, both AMT and DAIN experienced trajectory interruptions.

Our approach effectively captured this transition, demonstrating the robust modeling capabilities of the behavioral model for particle motion patterns. Quantitative tracking metrics indicated that our method produced more stable tracking results (Fig. 6b). Although DAIN outperformed our method in the CCR5 dataset, it exhibited a marked decline in performance on the Lysosome dataset, consistent with the earlier assessments of image generation quality.

3 Discussion

Video frame interpolation techniques can enhance the temporal resolution of live-cell imaging, facilitating the observation of rapid cellular dynamics while mitigating phototoxicity damage to samples. However, existing interpolation models struggle to effectively differentiate between visually similar subcellular components, resulting in significant shortcomings in interpolation outcomes. Our approach addresses this limitation by modeling the behavior of subcellular particles, embedding group behavioral features into individual appearance characteristics. Subcellular particles typically exhibit unique behaviors, which distinctly differentiate these microscopic entities from macroscopic objects. Thus, effectively leveraging this behavioral information serves as an excellent complement to the particles' uniqueness. We demonstrate that our approach is applicable across various representative subcellular particle behaviors and motion patterns, while also exhibiting stability in downstream particle tracking tasks.

The experimental results indicate that the behavioral feature excels in subcellular frame interpolation. In low-density simulated datasets, the behavioral feature consistently captures the appearance, disappearance, and rapid movement of small particles with high accuracy. In high-density scenarios, our approach delivers interpolation results that are closely aligned with ground truth images, even in regions where particle overlap occurs. Particle tracking experiments on the interpolated frames show significantly fewer trajectory interruptions and errors with our approach than with other methods. For real dataset interpolations, our approach maintains stability and demonstrates exceptional performance, particularly on the EB1 dataset, where its interpolated frames are the closest to the ground truth. Tracking experiments on these real datasets further show that our approach reduces common tracking errors, such as trajectory disappearance, compared to other state-of-the-art methods, highlighting its robustness and effectiveness in handling subcellular dynamics.

We outline two key capabilities that distinguish Sub-VFI from other frame interpolation models: (1) Robustness in challenging scenarios: Sub-VFI consistently outperforms other methods in difficult conditions, such as when particles appear or disappear, move rapidly, or overlap. These challenging situations are common in biological datasets, and our approach's ability to accurately interpolate under these conditions demonstrates its superior robustness. This robustness is primarily attributed to the accurate modeling of particle motion patterns by the behavioral model. (2) Enhanced generalization ability: The behavioral feature exhibits stable performance across datasets with diverse subcellular behaviors and motion patterns. In contrast, other methods often show performance degradation in certain datasets, indicating that relying solely on subcellular appearance features is insufficient for distinguishing and

matching subcellular components under these challenging scenarios. Our approach effectively mitigates this limitation, allowing for more accurate modeling of subcellular dynamics across different datasets and conditions. These capabilities highlight behavioral features’ unique advantage in subcellular frame interpolation, even in the most complex scenarios.

Subcellular image interpolation solutions hold great promise for enhancing microscopy bioimaging by reducing laser exposure or enabling higher frame rates to study rapid subcellular processes. Despite this potential, these solutions have limitations: First, as noted by CAFI[12], interpolation networks may introduce artifacts, and Sub-VFI can produce errors such as particle loss or overly smooth results. Therefore, it is critical to conduct domain-specific assessments and downstream task analyses before applying interpolated results in practical bioimaging applications[43]. Second, despite similar particle motion patterns, we observed a noticeable performance gap when transitioning from simulated to real datasets. Models fine-tuned on simulated data often underperform on real datasets, a challenge not unique to our approach but shared by other interpolation methods. Fine-tuning on real data is essential, which limits the generalization of our approach. Finally, subcellular images may contain additional interfering components or image noise, posing further challenges for subcellular image interpolation. In future work, addressing dataset issues will remain a primary focus; we aim to explore more generalized simulated datasets and a broader range of real datasets. Additionally, effective denoising of images will be one of the methods to enhance the quality of subcellular interpolation.

In summary, we explored the incorporation of an additional channel within subcellular images as supplementary input for learning subcellular dynamics and demonstrated that analyzing subcellular behavior is a crucial approach for post-processing subcellular images. This method not only improves the temporal resolution of subcellular images but also enhances the accuracy of subcellular dynamic modeling. We demonstrated the effectiveness of this approach across various datasets and further validated its robustness through particle tracking in downstream tasks. We hope that this exploration will inspire further advancements in the study of subcellular dynamics, paving the way for more accurate and efficient bioimaging techniques.

4 Methods

4.1 Dataset Preparation

Our simulated test data were sourced from the 2014 ISBI Particle Tracking Challenge[33], which provided three types of 3D (2D+T) particle images: Microtubules, Vesicles, and Receptors. These datasets correspond to directed motion, Brownian motion, and random switching between these two types of motion, respectively. We selected the low-density and high-density datasets from each category as test data, with each dataset comprising 100 images. To generate adequate training data, we employed the ISBI Challenge Tracking Benchmark Generator plugin within the ICY software[38], generating a total of 6,000 images (2,000 images at a time, repeated three times) for each particle type along with tracking ground truth files for

the image sequences. The parameters for particle simulation were set to default values to maintain consistency with the test data. All parameter values for the particle simulation datasets are detailed in Supplementary Table S1.

Using the tracking ground truth files, we extracted particle coordinates from each image, along with the particle-level flow between adjacent frames. These coordinates and flows will be used to train the subcellular behavior model and the interpolation network.

To assess the effectiveness of our method on real datasets across different subcellular behavior and particle motion patterns, we collected three real datasets from the literature, ensuring that their motion patterns directly correspond to those in the simulated datasets. EB1, exhibiting directed motion, was identified as an APC-binding protein and reported to be associated with microtubules[31]. CC chemokine receptor 5 (CCR5), primarily exhibiting Brownian motion, is a major receptor involved in inflammation and serves as the principal entry co-receptor for HIV-1[32]. Lysosomal behavior is a highly dynamic phenomenon that plays a crucial role in essential cellular processes such as degradation and repair mechanisms[12]. In the datasets we collected, its motion patterns switch between directed motion and Brownian motion. Further details about these datasets can be found in Supplementary Table S2.

4.2 Subcellular Behavior Model

The subcellular behavior model aims to characterize the behavior of particles within subcellular images, capturing their behavioral features. Inspired by scene flow methods, we input the coordinates of particles from different frames and output their matching features. Scene flow learning networks[44–47] are widely employed for motion estimation in point clouds, with common applications in autonomous driving and robotics. Recently, some researchers have begun to apply this type of network to Particle Tracking Velocimetry (PTV)[48, 49]. Here, we applied this approach to model subcellular behavior.

To effectively focus on the behavior of the entire particle population, it is essential that the behavioral features extracted by the model possess a global perspective. The GMSF[47] model is an excellent choice for this purpose, which employs a hybrid local-global-cross Transformer architecture to learn accurate and robust feature representations. The GMSF network has demonstrated strong capabilities across several scene flow estimation benchmarks, including FlyingThings3D[50] and KITTI[51] Scene Flow. GMSF begins with an off-the-shelf feature extraction network, DGCNN[52], and a local Point Transformer[53] for point cloud tokenization. It then utilizes self-attention and cross-attention operations to capture correlations among points within the same frame and across adjacent frames. After extracting features from both frames, GMSF computes a cross-similarity matrix to derive matching confidence, facilitating point cloud matching for initial scene flow estimation. A self-similarity matrix for the first frame is computed and multiplied by the initial scene flow, allowing for the propagation of scene flow from non-occluded to occluded regions, resulting in the final scene flow. We adapted the GMSF data format from three-dimensional to two-dimensional coordinates, feeding the particle coordinates detected in subcellular images into the model.

4.3 VFI Network Architectures

For the interpolation network, we selected the AMT[17] model, which employs a coarse-to-fine framework and utilizes all-pairs correlation in RAFT[54] to iteratively refine the bilateral backward optical flow. In the final iteration, multiple sets of optical flows are generated and input into the frame synthesis module, effectively mitigating ambiguities caused by occluded regions.

As the behavioral features extracted by the GMSF-based behavior model are organized according to the input order of particles, while the interpolation network requires features arranged in a two-dimensional grid corresponding to their spatial positions, a conversion is necessary. We map the structural features to a two-dimensional vector that aligns with the dimensions of the image features based on the particles' two-dimensional coordinates, subsequently fusing these behavioral features with the appearance features extracted by the appearance feature extractor.

4.4 Training Procedure

We employed a two-stage training strategy. In the first stage, the subcellular behavior model was trained using the simulated training dataset. Particle coordinates from adjacent images served as input, while the particle-level flow between these images acted as labels. For the six categories of simulated datasets—encompassing three distinct motion patterns at both high and low densities—a separate subcellular behavior model was trained for each category.

In the second stage, we trained the feature fusion module and the interpolation network. For the six distinct simulated datasets, we first loaded the corresponding pre-trained subcellular behavior model and froze its parameters. Subsequently, we copied the parameters from the pre-trained AMT model into our appearance feature extractor and frame synthesis module, followed by training on the simulated datasets. The input for this stage consists of particle coordinates and the first and third images from a sequence of three consecutive frames. The particle coordinates are provided to the subcellular behavior model to extract structural features, while the images are fed into the frame synthesis network to generate the intermediate frame. The second image in the sequence serves as the ground truth. For the real datasets, we loaded the subcellular behavior model according to their motion patterns and density levels, with the CCR5 and EB1 datasets classified as low density and the Lysosome dataset as high density.

4.5 Training Loss

The training losses in both stages were largely consistent with those used in GMSF and AMT, respectively.

For the first stage, let $\hat{V}$ denote the estimated particle-level scene flow and V_{gt} the ground truth; the training loss is defined as

$$\mathcal{L}_{\hat{V}} = \sum_i \left(\left\| \hat{V}(i) - V_{gt}(i) \right\|_1 + \epsilon \right)^q \quad (1)$$

where ϵ is set to 0.01 and q is set to 0.4. Both the final and intermediate scene flow estimation are used during training. The final loss is given as

$$\mathcal{L}_1 = \mathcal{L}_{\hat{V}_{final}} + \gamma \mathcal{L}_{\hat{V}_{inter}} \quad (2)$$

with γ set to 0.9.

In the second stage, we employed the Charbonnier loss[55] $\mathcal{L}_{char}$ and the census loss[56] $\mathcal{L}_{css}$ to supervise the generation of the interpolated frames, and the flow distillation loss $\mathcal{L}_{flow}$ to better predict task-oriented flows. The Charbonnier loss quantifies the pixel-wise difference between the ground truth intermediate frame and the generated frame, while the census loss computes the soft Hamming distance between census-transformed patches of and. The overall training loss for this stage is defined as follows:

$$\mathcal{L}_2 = \lambda_{char} \mathcal{L}_{char} + \lambda_{css} \mathcal{L}_{css} + \lambda_{flow} \mathcal{L}_{flow} \quad (3)$$

where λ_{char} , λ_{css} , and λ_{flow} are set as 1, 1, and 0.002.

4.6 Training details

Our model was implemented using PyTorch. For the first training phase, we used the AdamW optimizer with a weight decay of 10^{-4} and OneCycleLR as the scheduler to adjust the learning rate dynamically. The initial learning rate was set to 2×10^{-4} , and the batch size was set to 64. Other parameters can be found in Supplementary Table S7. For the second training phase, we used the AdamW optimizer without weight decay and the learning rate decay follows the cosine attenuation schedule from 2×10^{-4} to 2×10^{-5} . The batch size was set to 1, and training was conducted for 15 epochs.

4.7 Particle Tracking

Particle tracking was executed using the Fiji[42] plugin TrackMate[41]. For the detection of particles in the simulated datasets, we employed the Laplacian of Gaussian (LoG) detector. The linking process was carried out using the "Simple LAP tracker." Detailed parameters for detection and linking specific to each dataset are outlined in Supplementary Table S3.

For the simulated and real datasets, we implemented different experimental setups. For the simulated datasets, we have tracking ground truth files. For an image sequence of length n , the t and $t+2$ images are used as inputs, where t ranges from $[1, n-2]$ with an increment of 1. This allows us to obtain interpolation results for all intermediate frames, from the second to the $n-1$ frames, fully utilizing the tracking ground truth. We then combine these interpolation results with the first and last images of the sequence to create the final sequence, using TrackMate for tracking and comparing the tracking results with the ground truth through the ICY ISBI Challenge Tracking Batch Scoring plugin.

Since the real datasets lack tracking ground truth files, we utilize TrackMate to track the real image sequences and use the obtained tracking results as the ground truth for the tracking experiments. During interpolation, t is incremented by 2 each time, effectively downsampling the real image sequences and then using interpolation

484 methods for upsampling, allowing us to compare the accuracy of different interpolation
485 methods in enhancing the temporal resolution of the real image sequences.

486 4.8 Evaluation metrics

487 We utilized Peak Signal-to-Noise Ratio[40] (PSNR) and Structural Similarity Index[39]
488 (SSIM), which are standard metrics in frame interpolation tasks, to assess both pixel-
489 level and structural similarities between the interpolated images and the ground truth
490 images. For the tracking experiments, we adopted five performance criteria from the
491 ISBI Particle Tracking Challenge: Pairing distance, Normalized pairing score (alpha),
492 Full normalized score (beta), Similarity between tracks (Jaccard), and Similarity
493 between detections (Jaccard).

494 4.9 Competing methods

495 We compared our approach against several state-of-the-art interpolation methods,
496 including AMT[17], M2M[18], DAIN[15], and Zooming SlowMo[13, 14] (ZS). AMT
497 serves as the interpolation network model for our method; M2M is a classic for-
498 ward optical flow-based method that, like AMT, utilizes a multi-flow design; DAIN
499 and ZS are two models employed in CAFI[12], both recognized for their exceptional
500 interpolation performance.

501 To ensure a fair comparison, we downloaded the pre-trained models for AMT,
502 M2M, and DAIN from their respective GitHub repositories, and the pre-trained ZS
503 model, specifically tailored for enhancing temporal resolution, from the CAFI-provided
504 Colab code. Consistent with the second stage of training for our approach, we fine-
505 tuned all competing methods using simulated and real datasets individually. All
506 experiments were conducted on a uniform machine equipped with an Nvidia GeForce
507 RTX 2080Ti GPU.

508 Data availability

509 The simulated testing dataset (<http://bioimageanalysis.org/track/>) is available from
510 the 2014 ISBI Particle Tracking Challenge. The real datasets used in this study
511 were obtained from previous publications. Specifically, the EB1 dataset was sourced
512 from the paper published by Mimori-Kiyosue Y *et al.* in 2000[31], the CCR5
513 dataset from the paper published by Momboisse F *et al.* in 2022[32], and the
514 Lysosome dataset from the paper published by Priessner M *et al.* in 2024[12].
515 All training and testing datasets involved in the experiments, including the simu-
516 lated training datasets generated during this study, can be downloaded via Zenodo
517 at(<https://zenodo.org/records/14043236/>).

518 Code availability

519 The code is available online at <https://github.com/sduzzx857/SuB-VFI>.

Acknowledgements

This work was supported in part by NSFC under Grants U2001209 and 62372117 and in part by the Natural Science Foundation of Shanghai under Grant 21ZR1406600.

Competing interests

The authors declare no competing interests.

Supplementary information

Supplementary Figure S1 shows the architecture of our subcellular behavior-based VFI model.

Supplementary Figure S2 shows examples of the sudden appearance and disappearance of particles in the Microtubule dataset.

Supplementary Figure S3 shows visualizations of representative trajectories from real datasets.

Supplementary Table S1 shows parameters for creating simulated training datasets with the ICY plugin.

Supplementary Table S2 shows the overview of the real dataset details.

Supplementary Table S3 shows parameters for tracking experiments with Fiji Trackmate plugin.

Supplementary Table S4 shows the evaluation of behavior model on simulated datasets.

Supplementary Table S5 shows the PSNR/SSIM evaluation of DAIN, M2M, ZS, AMT, and Ours interpolation.

Supplementary Table S6 shows the tracking evaluation of DAIN, M2M, ZS, AMT, and Ours interpolation.

Supplementary Table S7 shows parameters for training behavior model.

References

- [1] Ma C, Tan W, He R, Yan B. Pretraining a foundation model for generalizable fluorescence microscopy-based image restoration. *Nature Methods*. 2024 Aug;21(8):1558–1567. <https://doi.org/10.1038/s41592-024-02244-3>.
- [2] Li X, et al. Spatial redundancy transformer for self-supervised fluorescence image denoising. *Nature Computational Science*. 2023 Dec;3(12):1067–1080. <https://doi.org/10.1038/s43588-023-00568-2>.
- [3] Bilodeau A, et al. Microscopy analysis neural network to solve detection, enumeration and segmentation from image-level annotations. *Nature Machine Intelligence*. 2022 May;4(5):455–466. <https://doi.org/10.1038/s42256-022-00472-w>.

- 554 [4] Weigert M, et al. Content-aware image restoration: pushing the limits of flu-
555 orescence microscopy. *Nature Methods*. 2018 Dec;15(12):1090–1097. <https://doi.org/10.1038/s41592-018-0216-7>.
556
- 557 [5] Zhang K, et al. An end-to-end recurrent compressed sensing method to denoise,
558 detect and demix calcium imaging data. *Nature Machine Intelligence*. 2024
559 Sep;6(9):1106–1118. <https://doi.org/10.1038/s42256-024-00892-w>.
- 560 [6] Wu Y, Shroff H. Faster, sharper, and deeper: structured illumination microscopy
561 for biological imaging. *Nature Methods*. 2018 Dec;15(12):1011–1019. <https://doi.org/10.1038/s41592-018-0211-z>.
562
- 563 [7] Wu Y, et al. Multiview confocal super-resolution microscopy. *Nature*. 2021
564 Dec;600(7888):279–284. <https://doi.org/10.1038/s41586-021-04110-0>.
- 565 [8] Kang I, et al. Coordinate-based neural representations for computational adaptive
566 optics in widefield microscopy. *Nature Machine Intelligence*. 2024 Jun;6(6):714–
567 725. <https://doi.org/10.1038/s42256-024-00853-3>.
- 568 [9] Bilodeau A, et al. Development of AI-assisted microscopy frameworks
569 through realistic simulation with pySTED. *Nature Machine Intelligence*. 2024
570 Oct;6(10):1197–1215. <https://doi.org/10.1038/s42256-024-00903-w>.
- 571 [10] Wang Z, et al. STDIN: Spatio-temporal distilled interpolation for electron
572 microscope images. *Neurocomputing*. 2022;505:188–202. <https://doi.org/https://doi.org/10.1016/j.neucom.2022.07.037>.
573
- 574 [11] Saguy A, et al. DBlink: dynamic localization microscopy in super spatiotemporal
575 resolution via deep learning. *Nature Methods*. 2023 07;20:1–10. <https://doi.org/10.1038/s41592-023-01966-0>.
576
- 577 [12] Priessner M, et al. Content-aware frame interpolation (CAFI): deep learning-
578 based temporal super-resolution for fast bioimaging. *Nature Methods*. 2024
579 Feb;21(2):322–330. <https://doi.org/10.1038/s41592-023-02138-w>.
- 580 [13] Xiang X, et al. Zooming Slow-Mo: Fast and Accurate One-Stage Space-Time
581 Video Super-Resolution. In: 2020 IEEE/CVF Conference on Computer Vision
582 and Pattern Recognition (CVPR); 2020. p. 3367–3376.
- 583 [14] Xiang X, et al. Zooming SlowMo: An Efficient One-Stage Framework for Space-
584 Time Video Super-Resolution. *CoRR*. 2021;abs/2104.07473. [2104.07473](https://arxiv.org/abs/2104.07473).
- 585 [15] Bao W, et al. Depth-Aware Video Frame Interpolation. In: 2019 IEEE/CVF
586 Conference on Computer Vision and Pattern Recognition (CVPR); 2019. p. 3698–
587 3707.

- 588 [16] Nah S, et al. AIM 2019 Challenge on Video Temporal Super-Resolution: Methods
589 and Results. In: 2019 IEEE/CVF International Conference on Computer Vision
590 Workshop (ICCVW); 2019. p. 3388–3398.
- 591 [17] Li Z, et al. AMT: All-Pairs Multi-Field Transforms for Efficient Frame Inter-
592 polation. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern
593 Recognition (CVPR); 2023. p. 9801–9810.
- 594 [18] Hu P, et al. Many-to-many Splatting for Efficient Video Frame Interpolation.
595 In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition
596 (CVPR); 2022. p. 3543–3552.
- 597 [19] Kong L, et al. IFRNet: Intermediate Feature Refine Network for Efficient Frame
598 Interpolation. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern
599 Recognition (CVPR); 2022. p. 1959–1968.
- 600 [20] Jiang H, et al. Super SloMo: High Quality Estimation of Multiple Intermediate
601 Frames for Video Interpolation. In: 2018 IEEE/CVF Conference on Computer
602 Vision and Pattern Recognition; 2018. p. 9000–9008.
- 603 [21] Lee H, et al. AdaCoF: Adaptive Collaboration of Flows for Video Frame Inter-
604 polation. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern
605 Recognition (CVPR); 2020. p. 5315–5324.
- 606 [22] Ding T, Liang L, Zhu Z, Zharkov I. CDFI: Compression-Driven Network Design
607 for Frame Interpolation. In: 2021 IEEE/CVF Conference on Computer Vision
608 and Pattern Recognition (CVPR); 2021. p. 7997–8007.
- 609 [23] Huang Z, et al. Real-Time Intermediate Flow Estimation for Video Frame Inter-
610 polation. In: Computer Vision – ECCV 2022. Cham: Springer Nature Switzerland;
611 2022. p. 624–642.
- 612 [24] Liu B, et al. A subpellicular microtubule dynein transport machinery
613 regulates ookinete morphogenesis for mosquito transmission of *Plasmodium*
614 *yoelii*. *Nature Communications*. 2024 Oct;15(1):8590. [https://doi.org/10.1038/
615 s41467-024-52970-7](https://doi.org/10.1038/s41467-024-52970-7).
- 616 [25] Chang S, et al. Speckle microscopic evaluation of microtubule transport in
617 growing nerve processes. *Nature Cell Biology*. 1999 Nov;1(7):399–403. [https:
618 //doi.org/10.1038/15629](https://doi.org/10.1038/15629).
- 619 [26] Saftig P, Klumperman J. Lysosome biogenesis and lysosomal membrane pro-
620 teins: trafficking meets function. *Nature Reviews Molecular Cell Biology*. 2009
621 Sep;10(9):623–635. <https://doi.org/10.1038/nrm2745>.

- [27] Ballabio A, Bonifacino JS. Lysosomes as dynamic regulators of cell and organismal homeostasis. *Nature Reviews Molecular Cell Biology*. 2020 Feb;21(2):101–118. <https://doi.org/10.1038/s41580-019-0185-4>.
- [28] Baudet S, et al. Subcellular second messenger networks drive distinct repellent-induced axon behaviors. *Nature Communications*. 2023 Jun;14(1):3809. <https://doi.org/10.1038/s41467-023-39516-z>.
- [29] Bertot L, et al. Quantitative and Statistical Study of the Dynamics of Clathrin-Dependent and -Independent Endocytosis Reveal a Differential Role of EndophilinA2. *Cell Reports*. 2018 Feb;22(6):1574–1588. <https://doi.org/10.1016/j.celrep.2018.01.039>.
- [30] Matov A, et al. Analysis of microtubule dynamic instability using a plus-end growth marker. *Nature Methods*. 2010 Sep;7(9):761–768. <https://doi.org/10.1038/nmeth.1493>.
- [31] Mimori-Kiyosue Y, Shiina N, Tsukita S. The dynamic behavior of the APC-binding protein EB1 on the distal ends of microtubules. *Current Biology*. 2000 Jul;10(14):865–868. [https://doi.org/10.1016/S0960-9822\(00\)00600-X](https://doi.org/10.1016/S0960-9822(00)00600-X).
- [32] Momboisse F, et al. Tracking receptor motions at the plasma membrane reveals distinct effects of ligands on CCR5 dynamics depending on its dimerization status. *eLife*. 2022 jul;11:e76281. <https://doi.org/10.7554/eLife.76281>.
- [33] Chenouard N, et al. Objective comparison of particle tracking methods. *Nature Methods*. 2014;11:281 – 289.
- [34] Jaqaman K, et al. Robust single-particle tracking in live-cell time-lapse sequences. *Nature Methods*. 2008 Aug;5(8):695–702. <https://doi.org/10.1038/nmeth.1237>.
- [35] Chenouard N, Bloch I, Olivo-Marin JC. Multiple Hypothesis Tracking for Cluttered Biological Image Sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2013;35(11):2736–3750. <https://doi.org/10.1109/TPAMI.2013.97>.
- [36] Spilger R, et al. A Recurrent Neural Network for Particle Tracking in Microscopy Images Using Future Information, Track Hypotheses, and Multiple Detections. *IEEE Transactions on Image Processing*. 2020;29:3681–3694. <https://doi.org/10.1109/TIP.2020.2964515>.
- [37] Rezatofighi SH, et al. Joint Probabilistic Data Association Revisited. In: 2015 IEEE International Conference on Computer Vision (ICCV); 2015. p. 3047–3055.
- [38] de Chaumont F, et al. Icy: an open bioimage informatics platform for extended reproducible research. *Nature Methods*. 2012;9:690–696.

- [39] Wang Z, et al. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*. 2004;13(4):600–612. <https://doi.org/10.1109/TIP.2003.819861>.
- [40] Krull A, Buchholz TO, Jug F. Noise2Void - Learning Denoising From Single Noisy Images. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2019. p. 2124–2132.
- [41] Tinevez JY, et al. TrackMate: An open and extensible platform for single-particle tracking. *Methods*. 2017;115:80–90. *Image Processing for Biologists*. <https://doi.org/https://doi.org/10.1016/j.ymeth.2016.09.016>.
- [42] Schindelin JE, et al. Fiji: an open-source platform for biological-image analysis. *Nature Methods*. 2012;9:676–682.
- [43] Laine RF, et al. Avoiding a replication crisis in deep-learning-based bioimage analysis. *Nature Methods*. 2021 Oct;18(10):1136–1144. <https://doi.org/10.1038/s41592-021-01284-3>.
- [44] Liu X, Qi CR, Guibas LJ. FlowNet3D: Learning Scene Flow in 3D Point Clouds. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2019. p. 529–537.
- [45] Puy G, Boulch A, Marlet R. FLOT: Scene Flow on Point Clouds Guided by Optimal Transport. In: *Computer Vision – ECCV 2020*. Cham: Springer International Publishing; 2020. p. 527–544.
- [46] Wei Y, et al. PV-RAFT: Point-Voxel Correlation Fields for Scene Flow Estimation of Point Clouds. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; 2021. p. 6954–6963.
- [47] Zhang Y, et al. GMSF: Global Matching Scene Flow. In: Oh A, Naumann T, Globerson A, Saenko K, Hardt M, Levine S, editors. *Advances in Neural Information Processing Systems*. vol. 36. Curran Associates, Inc.; 2023. p. 64415–64427. Available from: https://proceedings.neurips.cc/paper_files/paper/2023/file/cb1c4782f159b55380b4584671c4fd88-Paper-Conference.pdf.
- [48] Liang J, et al. DeepPTV: Particle Tracking Velocimetry for Complex Flow Motion via Deep Neural Networks. *IEEE Transactions on Instrumentation and Measurement*. 2022;71:1–16. <https://doi.org/10.1109/TIM.2021.3120127>.
- [49] Liang J, Xu C, Cai S. Recurrent graph optimal transport for learning 3D flow motion in particle tracking. *Nature Machine Intelligence*. 2023 May;5(5):505–517. <https://doi.org/10.1038/s42256-023-00648-y>.
- [50] Mayer N, Ilg E, Hausser P, Fischer P, Cremers D, Dosovitskiy A, et al. A Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene

- 693 Flow Estimation. In: Proceedings of the IEEE Conference on Computer Vision
694 and Pattern Recognition (CVPR); 2016. .
- 695 [51] Menze M, Geiger A. Object scene flow for autonomous vehicles. In: 2015 IEEE
696 Conference on Computer Vision and Pattern Recognition (CVPR); 2015. p. 3061–
697 3070.
- 698 [52] Wang Y, et al. Dynamic Graph CNN for Learning on Point Clouds. ACM
699 Transactions on Graphics (TOG). 2018;38:1 – 12.
- 700 [53] Zhao H, et al. Point Transformer. In: 2021 IEEE/CVF International Conference
701 on Computer Vision (ICCV); 2021. p. 16239–16248.
- 702 [54] Teed Z, Deng J. RAFT: Recurrent All-Pairs Field Transforms for Optical Flow.
703 In: Computer Vision – ECCV 2020. Cham: Springer International Publishing;
704 2020. p. 402–419.
- 705 [55] Charbonnier P, et al. Two deterministic half-quadratic regularization algorithms
706 for computed imaging. In: Proceedings of 1st International Conference on Image
707 Processing. vol. 2; 1994. p. 168–172 vol.2.
- 708 [56] Meister S, Hur J, Roth S. UnFlow: Unsupervised Learning of Optical Flow with
709 a Bidirectional Census Loss. In: AAAI Conference on Artificial Intelligence; 2017.
710 Available from: <https://api.semanticscholar.org/CorpusID:19160323>.

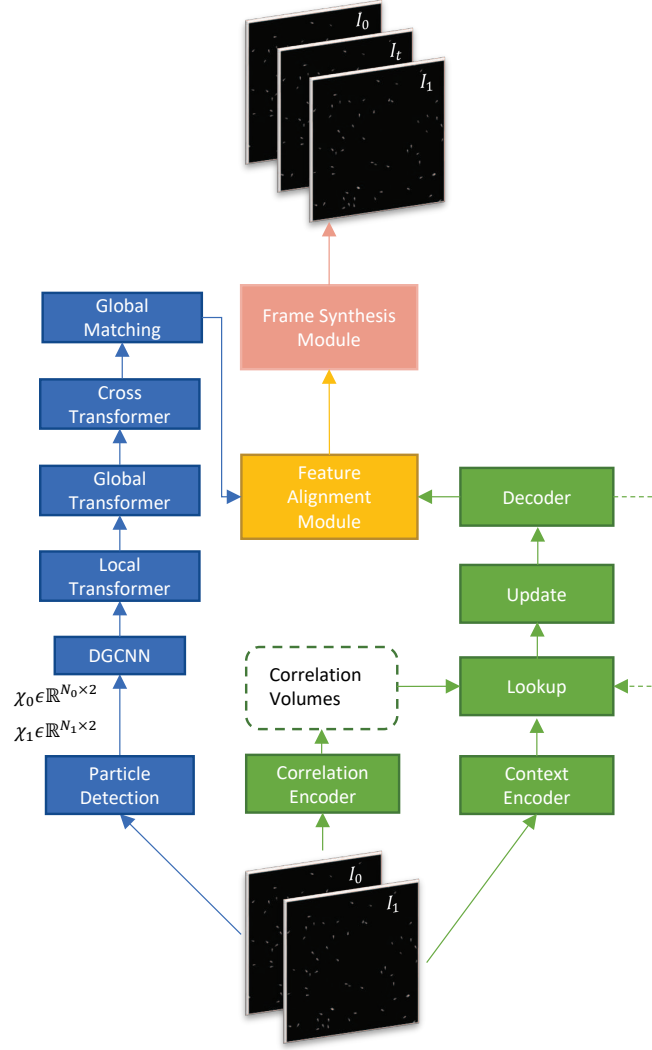


Fig. S1 Overall architecture of our subcellular behavior-based VFI model. The blue section represents the behavior model based on GMSF, while the green section represents the interpolation network based on AMT.

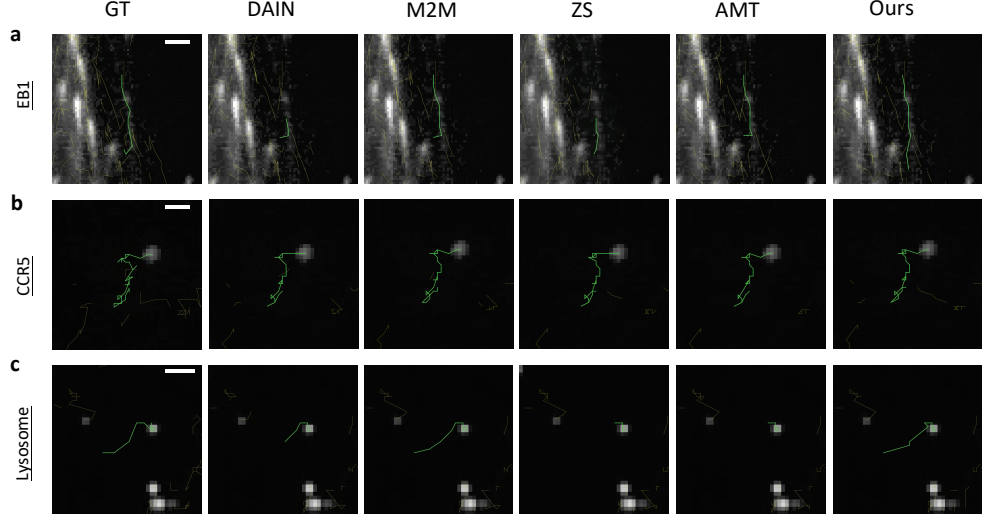


Fig. S2 Examples of sudden appearance and disappearance of particles in the Microtubule dataset. In the white boxed region of the I_0 , ground truth (GT), and I_1 , a particle suddenly appears in the upper right corner, while another particle suddenly disappears in the lower right corner. ZS and AMT fail to recover these two particles, and M2M demonstrates poor restoration, whereas DAIN and our method are closer to the ground truth.

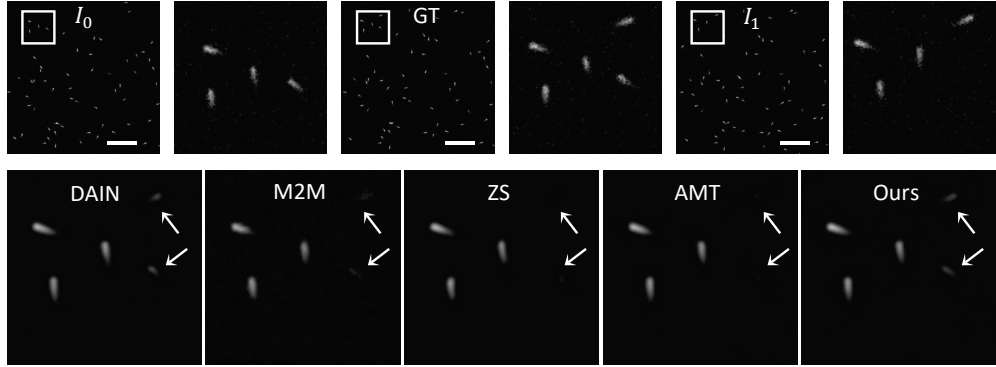


Fig. S3 Visualizations of representative trajectories from real datasets. **a**, The trajectories of particles performing directed motion along the y-axis in the EB1 dataset reveal that DAIN, M2M, and AMT display delayed appearance of trajectories, with DAIN and ZS experiencing premature interruptions. **b**, In the CCR5 dataset, the differences among various methods are minimal. **c**, During the Brownian motion phase in the Lysosome dataset, the methods show little variation; however, in the subsequent directed motion phase, trajectories from DAIN, ZS, and AMT exhibit interruptions.

ICY (ISBI Challenge Tracking Benchmark Generator)			
dataset	Microtubule	Vesicle	Receptor
image width	512	512	512
image height	512	512	512
sequence length	2000	2000	2000
SNR	7	7	7
density low	67	100	100
density high	760	1000	1000
image depth	1	1	1

Table S1 Parameters for creating simulated training datasets with ICY plugin.

Dataset	Microscopy	Imaging Organism	Publication	Training Data Number (Size)	Test
EB1	fluorescence	Xenopus A6 cells	[31]	34 (288×352)	34 (288×352)
CCR5	TIRF	HEK 293 cells	[32]	50 (496×496)	50 (496×496)
Lysosome	confocal	SH-SY5Y cells	[12]	20 (512×512)	19 (512×512)

Table S2 Overview of the real dataset details.

Trackmate Parameters	Microtubule	Vesicle	Receptor	EB1	CCR5	Lysosome
Estimated blob diameter	8	3	3	7	3	3
Threshold	3	12.5	15	5	15	10
Selected Tracker	Simple LAP tracker					
Linking max distance	8	8	8	8	8	8
Gap-closing max distance	8	8	8	8	8	8
Gap-closing max frame gap	2	2	2	2	2	2

Table S3 Parameters for tracking experiments with Fiji Trackmate plugin.

	$EPE_{2D}\downarrow$	$ACC_S\uparrow$	$ACC_R\uparrow$	$Outliers\downarrow$
Microtubule-low	0.0082	98.8829	98.9067	1.0933
Microtubule-high	0.0891	87.0842	87.3414	12.6586
Vesicle-low	0.0025	99.3416	99.3490	0.6285
Vesicle-high	0.0214	94.65	94.86	5.10
Receptor-low	0.0020	99.2378	99.2562	0.5693
Receptor-high	0.0158	95.9929	96.1117	3.6508

Table S4 Evaluation of behavior model on simulated datasets.

$EPE_{2D}\downarrow$ is the 2D endpoint error $\|\hat{V} - V_{gt}(i)\|_2$ between the estimated particle-level flow and the ground truth averaged over each particle. $ACC_S\uparrow$ is the percentage of the estimated particle-level flow with a relative error of less than 5%. $ACC_R\uparrow$ is the percentage of the estimated particle-level flow with a relative error of less than 10%. $Outliers\downarrow$ is the percentage of the estimated particle-level flow with a relative error of more than 10%.

	DAIN	M2M	ZS	AMT	Ours
Microtubule-low	35.55/0.7897	35.08/0.7809	35.35/0.7883	35.45/0.7891	35.56/0.7898
Vesicle-low	35.05/0.7976	34.96/0.7927	35.03/0.7968	<u>35.11/0.7980</u>	<u>35.10/0.7981</u>
Receptor-low	35.69/0.7907	35.43/0.7838	35.63/0.7899	35.71/0.7909	<u>35.71/0.7908</u>
Microtubule-high	32.95/0.8292	28.99/0.7278	32.56/0.8218	<u>33.06/0.8325</u>	<u>33.34/0.8363</u>
Vesicle-high	29.22/0.6490	29.05/0.6368	29.00/0.6423	<u>29.29/0.6503</u>	<u>29.27/0.6504</u>
Receptor-high	33.35/0.8077	32.93/0.7956	33.23/0.8058	<u>33.69/0.8150</u>	<u>33.59/0.8138</u>
EB1	29.22/0.6778	28.01/0.6119	27.40/0.6180	<u>28.78/0.6344</u>	<u>29.36/0.6860</u>
CCR5	41.70/ <u>0.9617</u>	41.51/0.9601	41.48/0.9600	<u>41.73/0.9614</u>	<u>41.69/0.9615</u>
Lysosome	<u>34.59/0.9550</u>	34.39/0.9610	34.47/0.9546	34.55/0.9536	34.56/ <u>0.9612</u>

Table S5 PSNR $\uparrow$ /SSIM $\uparrow$ evaluation of DAIN, M2M, ZS, AMT, and Ours interpolation. The best result is shown in underlined.

	DAIN	M2M	ZS	AMT	Ours
Microtubule	16396	21018	17319	16686	<u>16124</u>
	/0.4898/0.4707 /0.8875/0.7131	/0.3460/0.3209 /0.7119/0.5102	/0.4611/0.4420 /0.8653/0.6744	/0.4808/0.4610 /0.8883/0.6971	/0.4983/0.4831 <u>/0.8986/0.7299</u>
Vesicle	24453	28762	24538	24632	<u>24310</u>
	/0.5441/0.5278 /0.9267/0.8118	/0.4638/0.4260 /0.7782/0.6655	/0.5425/0.5271 <u>/0.9283/0.8123</u>	/0.5408/0.5224 /0.9128/0.8027	/0.5468/0.5294 <u>/0.9268/0.8134</u>
Receptor	13450	19071	13730	13881	<u>13108</u>
	/0.6383/0.6112 /0.8831/0.7313	/0.4871/0.4441 /0.7275/0.5332	/0.6307/0.5992 /0.8857/0.7269	/0.6267/0.5979 /0.8736/0.7173	/0.6475/0.6253 <u>/0.8980/0.7518</u>
EB1	5173	5940	6480	5215	<u>4806</u>
	/0.3254/0.2701 /0.4801/0.3385	/0.2255/0.1794 /0.3382/0.2214	/0.1551/0.1226 /0.2617/0.1683	/0.3200/0.2766 /0.4516/0.3264	/0.3733/0.3185 <u>/0.4947/0.3673</u>
CCR5	6348	6750	7198	6800	6713
	<u>/0.5929/0.5700</u> <u>/0.3878/0.6254</u>	/0.5671/0.5312 /0.3605/0.5836	/0.5384/0.5130 /0.3076/0.5596	/0.5639/0.5338 /0.3555/0.5837	/0.5695/0.5392 /0.3633/0.5876
Lysosome	15008	14768	15155	14579	<u>14333</u>
	/0.6559/0.6120 /0.6145/0.6628	/0.6614/0.6194 /0.6143/0.6694	/0.6525/0.6145 /0.625/0.6699	/0.6657/0.6234 <u>/0.6346/0.6718</u>	<u>/0.6714/0.6264</u> <u>/0.6346/0.6792</u>

Table S6 Tracking evaluation of DAIN, M2M, ZS, AMT, and Ours interpolation. Five performance criteria from the ISBI Particle Tracking Challenge: Pairing distance↓, Normalized pairing score (alpha)↑, Full normalized score (beta)↑, Similarity between tracks (Jaccard)↑, and Similarity between detections (Jaccard)↑. The best result is shown in underlined.

Parameters	Microtubule low	Microtubule high	Vesicle low	Vesicle high	Receptor low	Receptor high
num_transformer_pt_layers	1	1	1	1	1	1
num_transformer_layers	4	4	4	4	4	4
feature_channels	64	64	64	64	64	64
npoints	85	750	135	1100	110	910
num_steps	20000	40000	20000	20000	20000	20000

Table S7 Parameters for training behavior model.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- [demo.mp4](#)